C interfaces to GALAHAD LHS

Jari Fowkes and Nick Gould
STFC Rutherford Appleton Laboratory
Thu Jun 22 2023

Chapter 1

GALAHAD C package lhs

1.1 Introduction

1.1.1 Purpose

This package **computes an array of Latin Hypercube samples.**.

Currently, only the control and inform parameters are exposed; these are provided and used by other GALAHAD packages with C interfaces.

1.1.2 Authors

J. Burkardt, University of Pittsburgh (LGPL) adapted for GALAHAD by N. I. M. Gould, STFC-Rutherford Appleton Laboratory, England.

C interface, additionally J. Fowkes, STFC-Rutherford Appleton Laboratory.

Julia interface, additionally A. Montoison and D. Orban, Polytechnique Montréal.

1.1.3 Originally released

June 2016, C interface March 2022.

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

File Documentation

3.1 galahad_lhs.h File Reference

```
#include <stdbool.h>
#include <stdint.h>
#include "galahad_precision.h"
#include "galahad_cfunctions.h"
```

Data Structures

- struct lhs_control_type
- struct lhs_inform_type

Functions

- void lhs_initialize (void **data, struct lhs_control_type *control, struct lhs_inform_type *inform)
- void lhs_read_specfile (struct lhs_control_type *control, const char specfile[ ])
- void lhs_ihs (int n_dimen, int n_points, int *seed, int **X, const struct lhs_control_type *control, struct lhs_inform_type *inform, void **data)
- void lhs_get_seed (int *seed)
- void lhs_information (void **data, struct lhs_inform_type *inform, int *status)
- void lhs_terminate (void **data, struct lhs_control_type *control, struct lhs_inform_type *inform)

3.1.1 Data Structure Documentation

3.1.1.1 struct lhs_control_type

Data Fields

int	error	error and warning diagnostics occur on stream error.
int	out	general output occurs on stream out.

Data Fields

int	print_level	the level of output required. Possible values are: • $<$ 1 no output. • $>$ 0 debugging.
int	duplication	the duplication factor. This must be at least 1, a value of 5 is reasonable.
bool	space_critical	if .space_critical true, every effort will be made to use as little space as possible. This may result in longer computation time.
bool	deallocate_error_fatal	if .deallocate_error_fatal is true, any array/pointer deallocation error will terminate execution. Otherwise, computation will continue.
char	prefix[31]	all output lines will be prefixed by .prefix(2:LEN(TRIM(prefix))-1) where .prefix contains the required string enclosed in quotes, e.g. "string" or 'string'

3.1.1.2 struct lhs_inform_type

Data Fields

int	status	return status. Possible values are: • 0 the call was successful. • -1. An allocation error occurred. A message indicating the offending array is written on unit control.error, and the returned allocation status and a string containing the name of the offending array are held in inform.alloc_status and inform.bad_alloc respectively. • -2. A deallocation error occurred. A message indicating the offending array is written on unit control.error and the returned allocation status and a string containing the name of the offending array are held in inform.alloc_status and inform.bad_alloc respectively. • -3. The random number seed has not been set.
int	alloc_status	the status of the last attempted allocation/deallocation.
char	bad_alloc[81]	the name of the array for which an allocation/deallocation error occurred.

3.1.2 Function Documentation

3.1.2.1 lhs_initialize()

```
void lhs_initialize (
             void ** data,
             struct lhs_control_type * control,
             struct lhs_inform_type * inform )
```

Set default control values and initialize private data

Parameters

in,out	*data*	holds private internal data
out	*control*	is a struct containing control information (see fit_control_type)
out	*inform*	is a struct containing output information (see fit_inform_type)

3.1.2.2 lhs_read_specfile()

```
void lhs_read_specfile (
             struct lhs_control_type * control,
             const char specfile[] )
```

Read the content of a specification file, and perform the assignment of values associated with given keywords to the corresponding control parameters.

By default, the spcification file will be named RUNLHS.SPC and lie in the current directory. Refer to Table 2.1 in the fortran documentation provided in $GALAHAD/doc/lhs.pdf for a list of keywords that may be set.

Parameters

in	*control*	a struct containing control information (see above)
in	*specfile*	a character string containing the name of the specfile

3.1.2.3 lhs_ihs()

```
void lhs_ihs (
             int n_dimen,
             int n_points,
             int * seed,
             int ** X,
             const struct lhs_control_type * control,
             struct lhs_inform_type * inform,
             void ** data )
```

The improved distributed hyper-cube sampling algorithm.

Discussion:

n_points points in an n_dimen dimensional Latin hyper-cube are to be selected. Each of the coordinate dimensions is discretized to the values 1 through n. The points are to be chosen in such a way that no two points have any coordinate value in common. This is a standard Latin hypercube requirement, and there are many solutions.

This algorithm differs in that it tries to pick a solution which has the property that the points are "spread out" as evenly as possible. It does this by determining an optimal even spacing, and using the DUPLICATION factor to allow it to choose the best of the various options available to it.

Reference:

Brian Beachkofski, Ramana Grandhi, Improved Distributed Hypercube Sampling, American Institute of Aeronautics and Astronautics Paper 2002-1274

Parameters

in	*n_dimen*	is a scalar variable of type int that specifies the spatial dimension
in	*n_points*	is a scalar variable of type int that specifies the number of points to be generated
in,out	*seed*	is a scalar variable of type int, that gives a seed for the random number generator used
out	*X*	is an array variable of type int with dimensions [n_dimen][n_points] that gives the hyper-cube points
in,out	*control,inform,data*	- see lhs_initialize

3.1.2.4 lhs_get_seed()

```
void lhs_get_seed (
            int * seed )
```

Get a seed for the random number generator.

Parameters

out	*seed*	is a scalar variable of type int that gives the pseudorandom seed value.

3.1.2.5 lhs_information()

```
void lhs_information (
            void ** data,
            struct lhs_inform_type * inform,
            int * status )
```

Provides output information

Parameters

in,out	*data*	holds private internal data
out	*inform*	is a struct containing output information (see lhs_inform_type)
out	*status*	is a scalar variable of type int, that gives the exit status from the package. Possible values are (currently): • 0. The values were recorded succesfully

3.1.2.6 lhs_terminate()

```
void lhs_terminate (
            void ** data,
            struct lhs_control_type * control,
            struct lhs_inform_type * inform )
```

Deallocate all internal private storage

Parameters

in,out	*data*	holds private internal data
out	*control*	is a struct containing control information (see lhs_control_type)
out	*inform*	is a struct containing output information (see lhs_inform_type)